

Doc Code: AP.PRE.REQ

PTO/SB/33 (07-09)

Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PRE-APPEAL BRIEF REQUEST FOR REVIEW		Docket Number (Optional) MWS-042RCE3	
	Application Number 10/057,126-Conf. #1888	Filed October 25, 2001	
	First Named Inventor Kai TUSCHNER <i>et al.</i>		
	Art Unit 2178	Examiner A. L. Basehoar	
<p>Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.</p> <p>This request is being filed with a notice of appeal.</p> <p>The review is requested for the reason(s) stated on the attached sheet(s). Note: No more than five (5) pages may be provided.</p> <p>I am the</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <p><input type="checkbox"/> applicant /inventor.</p> <p><input type="checkbox"/> assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)</p> <p><input checked="" type="checkbox"/> attorney or agent of record. Registration number <u>35,470</u></p> <p><input type="checkbox"/> attorney or agent acting under 37 CFR 1.34. Registration number if acting under 37 CFR 1.34. _____</p> </div> <div style="width: 35%; text-align: center;"> <p>_____ /Kevin J. Canning/ Signature</p> <p>_____ Kevin J. Canning Typed or printed name</p> <p>_____ (617) 994-0732 Telephone number</p> <p>_____ March 4, 2010 Date</p> </div> </div> <p>NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.</p>			
<input type="checkbox"/> *Total of <u>1</u> forms are submitted.			

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being transmitted via the Office electronic filing system in accordance with § 1.6(a)(4).	
Dated: March 4, 2010	Electronic Signature for Kevin J. Canning: /Kevin J. Canning/

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being transmitted via the Office electronic filing system in accordance with § 1.6(a)(4).

Dated: March 4, 2010
Electronic Signature for Kevin J. Canning: / Kevin J. Canning/

Docket No.: MWS-042RCE3
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Kai Tuschner *et al.*

Application No.: 10/057,126

Confirmation No.: 1888

Filed: October 25, 2001

Art Unit: 2178

For: LINKED CODE GENERATION REPORT

Examiner: Adam L. Basehoar

ARGUMENTS FOR PRE-APPEAL BRIEF REQUEST FOR REVIEW

MS AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

The following is submitted together with a Notice of Appeal under 37 C.F.R. §41.31 and in support of a Pre-Appeal Brief Request for Review in the above-identified Application. Claims 1-3, 5-12, and 14-29 stand rejected under 35 U.S.C. § 103(c) as being unpatentable over Aptus (U.S. Patent No. 7,114,149) in view of Dori (U.S. Patent No. 6,961,686), and further in view of Kodosky (U.S. Patent No. 6,961,686). Applicants have appealed these rejections.

At issue in this appeal is a new capability made available in source code generators for block diagram models. Specifically, the present application addresses the production of a code generation report that provides a link between a portion of source code and an element in a block diagram model corresponding to the portion of source code.

Block diagrams are often used to model and simulate real-world systems, such as braking systems, shock absorbers, and electrical circuits (Application at page 1). For example, Figure 5 of the present Application depicts an exemplary block diagram model 193. Source code implementing the functionality of the block diagram model may be generated from the model (Application at pages 2-3).

However, it can be difficult to debug source code generated from block diagram models. The model diagram is represented in a source model language such as Simulink of the MathWorks, Inc. of Natick, Massachusetts (Application at pages 7-8). Code generation involves

converting the source model language into program source code such as C, Ada, Basic, and Java (Application at page 8). When the source model language is translated into program source code, it may be difficult or impossible to discern which blocks in the model gave rise to specific portions of the program source code.

Accordingly, the present application provides a code generation report, which is produced from source code generated from a block diagram model. The code generation report includes hyperlinks that link from a portion of the generated source code to an element in the block diagram model that gave rise to the portion of source code.

Using the methods described in the present Specification, it is possible to identify which elements of the block diagram model gave rise to specific portions of the generated source code, thereby greatly simplifying the debugging process. Doing so is neither disclosed nor rendered obvious by the combination of Aptus, Dori, and Kodosky.

Representative claim 1 recites, among other things:

generating source code from a simulatable block diagram model using a code compiler, the simulatable block diagram model represented in a source model language, ***wherein the generated source code includes one or more comments that include a block path identifying a section of the source model language that corresponds to an element in the block diagram model;***

generating a code generation report from the generated source code using a report compiler, the generating of the code generation report comprising:

parsing, using the report compiler, the one or more comments in the generated source code,

identifying, using the report compiler, the block path in the one or more comments, and

converting, using the report compiler, the generated source code into the code generation report by replacing the block path with a hypertext link that refers to the element of the block diagram model that corresponds to the section of the source model language identified by the block path, the hypertext link providing a hyperlink from the code generation report to the element in the block diagram model; and displaying the code generation report to a user.

The other independent claims 10, 19, 23, and 27 also recite the features of “at least one comment that includes a block path, the block path identifying a section of the source model language that corresponds to a block in the block diagram model” and generating a code generation report by replacing the comment with a link to a corresponding section of the source model language identified by the block path.

None of the references disclose or suggest generated source code having *one or more comments that include a block path identifying a section of the source model language that corresponds to an element in the block diagram model*, nor *converting the generated source code into the code generation report by replacing the block path with a hypertext link that refers to the element of the block diagram model that corresponds to the section of the source model language identified by the block path*. For purposes of this preappeal brief, Applicants focus on the concept of a *block path identifying a section of the source model language that corresponds to an element in the block diagram model*, as this feature is clearly missing from the references.

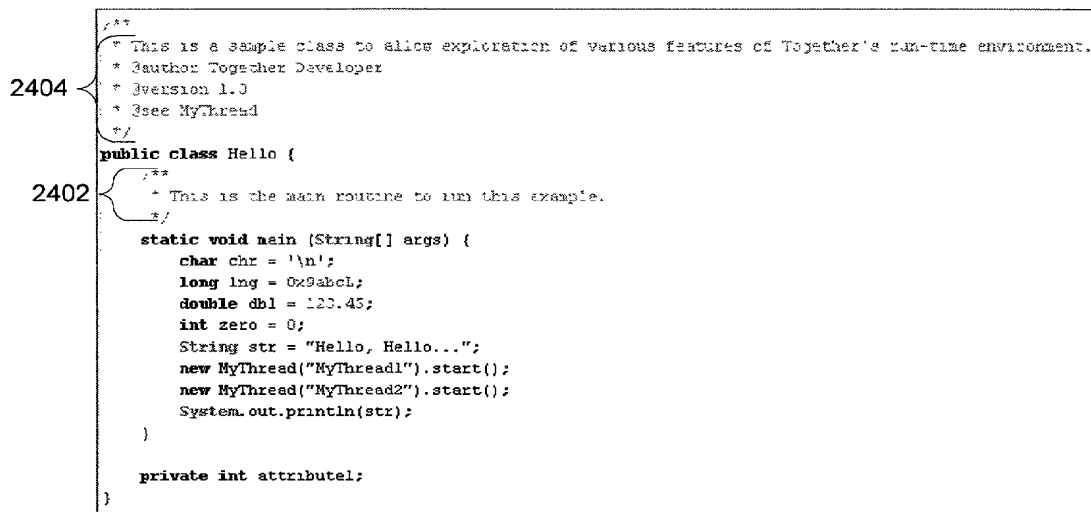
The Examiner does not suggest that any of Aptus, Dori, or Kodosky disclose or suggest source code having one or more comments including a *block path identifying a section of the source model language that corresponds to an element in the block diagram model*. Instead, the Examiner asserts that the term *block path* “appears to merely define a reference or identifier to a given object. Given its broadest reasonable interpretation, the block path of the claimed invention could be simply a name, reference, identification, or any other indicator of an object/element” (12/8/09 Office Action at page 11, paragraph 4).

The Examiner then asserts that Aptus teaches a block path because Aptus associates “corresponding portions of diagrammatic models and textual documentation based on the analysis of tokenized source code, which includes keywords, operators, punctuation, names, comments, parameters, etc” (12/8/09 Office Action at pages 11-12).

The Examiner is required to give claim terms their broadest reasonable interpretation. However, the Examiner is not entitled to ignore elements explicitly recited in the claims, nor is the Examiner entitled to distill the invention down to a mere “gist” or “thrust”(MPEP at §2142.01.II). The Examiner asserts that a block path may be any identifier such as a name or reference, apparently without regard to what the identifier identifies (Office Action at page 11, paragraph 4). However, the claim explicitly recites a type of identifier, namely a *block path*, as well as what the identifier does, namely *identifies a section of the source model language that corresponds to an element in the block diagram model*.

In Aptus, Examiner points to the “see” parameter in the source code as the equivalent of a block path (12/8/09 Office Action at page 3). However, the “see” parameter does not identify *a section of the source model language that corresponds to an element in the block diagram*

model as required by the claims. The “see” parameter is used in a comment describing the public class “Hello” to note that class “Hello” uses the MyThread function (Aptus at Figure 24):



```

2404  /**
2405   * This is a sample class to allow exploration of various features of Together's run-time environment.
2406   * @author Together Developer
2407   * @version 1.0
2408   * @see MyThread
2409   */
2410  public class Hello {
2411      /**
2412       * This is the main routine to run this example.
2413       */
2414      static void main (String[] args) {
2415          char chr = '\n';
2416          long lng = 0x9abcl;
2417          double dbl = 123.45;
2418          int zero = 0;
2419          String str = "Hello, Hello...";
2420          new MyThread("MyThread1").start();
2421          new MyThread("MyThread2").start();
2422          System.out.println(str);
2423      }
2424      private int attribute1;
2425  }

```

The “see” parameter does not identify *a section of the source model language that corresponds to an element in the block diagram model*, as required by claim 1. Instead, the “see” parameter is placed in the source code to identify the name of a function in the same source code. The present application places a block path in the source code comments that refers to a section of the source model language describing the block diagram model from which the source code was generated. In contrast, Aptus places an internal reference into source code that links between a class in the source code and a function in the source code.

The Examiner does not suggest that Dori or Kodosky discloses *a block path identifying a section of the source model language that corresponds to an element in the block diagram model*. Indeed, neither Dori nor Kodosky disclose or suggest such a block path (8/12/09 Response at page 10).

At page 12 of the present Office Action, the Examiner further asserts that “hyperlinking two elements based on a block path is notoriously well known in the art.” Applicants respectfully disagree. While hyperlinking two elements has certainly been accomplished in the prior art, Applicants are not aware of any prior art that hyperlinks a block diagram model with source code based on *a block path identifying a section of the source model language that corresponds to an element in the block diagram model*.

The Examiner goes on to state that “aside from what is described above, the newly cited Horowitz and Lin references also clearly teach two different methods for hyperlinking between two reports, documents, and/or elements based on identified terms, tags, elements, block paths, etc.” (12/8/09 Office Action at page 12). Once again, the Examiner has impermissibly ignored the claim language and distilled the invention down to a “gist” or “thrust.” Neither the cited Horowitz nor the cited Lin reference discloses or suggests ***a block path identifying a section of the source model language that corresponds to an element in the block diagram model.***

Horowitz generally concerns providing hyperlinks between a source document and a target document. In Horowitz, a computer analyzes the source document and selects terms that appear to be relevant to a user based on the terms’ frequency of occurrence (Horowitz at Abstract). A target document is selected by identifying topics associated with, or described by, the selected terms (Horowitz at Abstract). Links are created between the selected terms and the target document using a URL (Horowitz at Abstract).

Lin is generally directed to a method for building a hypertext-based outline for a report (Lin at Abstract). A report definition specifying the contents and form of the report is received and evaluated (Lin at Abstract). The system determines if any report definition statements contained in the report definition specify that a hypertext-based outline entry is to be generated (Lin at Abstract). If so, outline entries are generated that provide a link to a location specified by the report definition statement (Lin at Abstract).

However, neither Horowitz nor Lin is concerned with block diagrams, source model languages, and source code generation. Indeed, each of the cited references are entirely silent with respect to ***a block path identifying a section of the source model language that corresponds to an element in the block diagram model.*** Accordingly, Applicants respectfully request withdrawal of the rejections and the allowance of all claims.

Dated: March 4, 2010

Respectfully submitted,

Electronic signature: /Kevin J. Canning/
Kevin J. Canning
Registration No.: 35,470
LAHIVE & COCKFIELD, LLP
One Post Office Square
Boston, Massachusetts 02109-2127
Attorney/Agent For Applicant